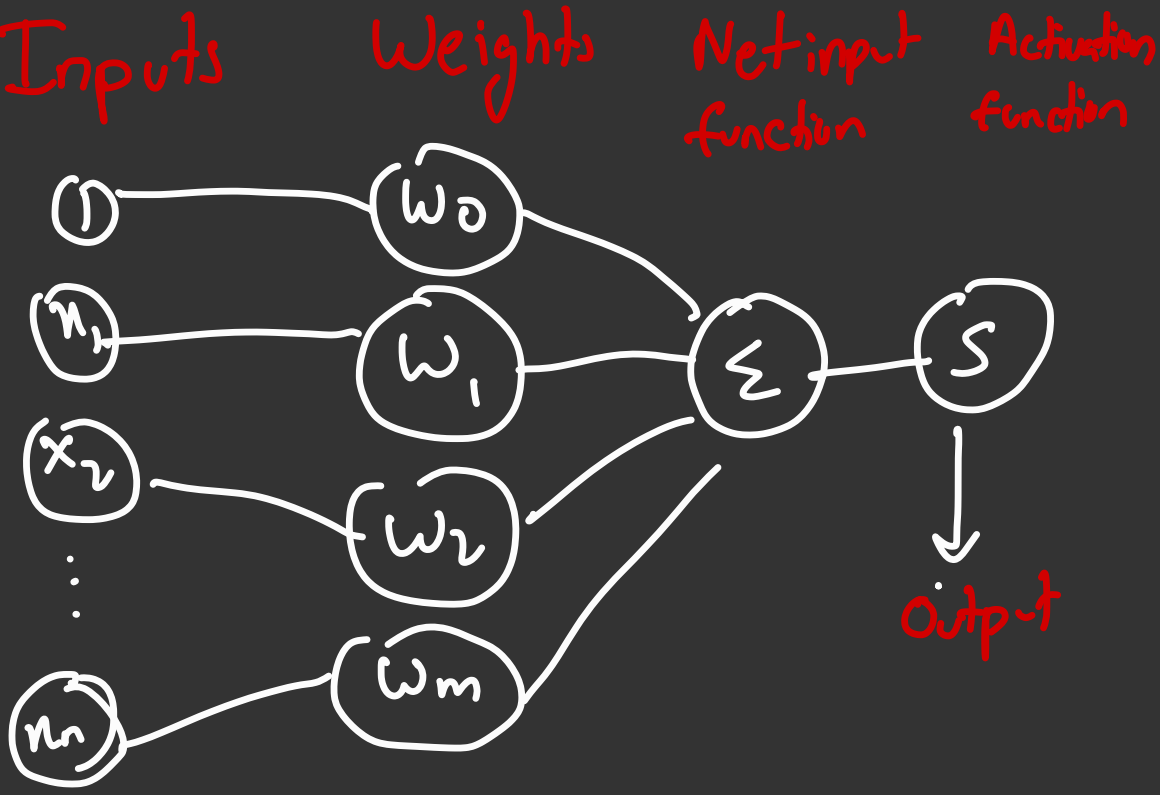
Perceptron from scratch:

- Similar to how our body recievs inputs like hearing and smelling and sends the reaction via neurons, Neural networks work in the same way
- A perceptron mimics the function of 1 neuron



Linear model

$$f(w,b) = w^T \cdot x + b$$

Activation function

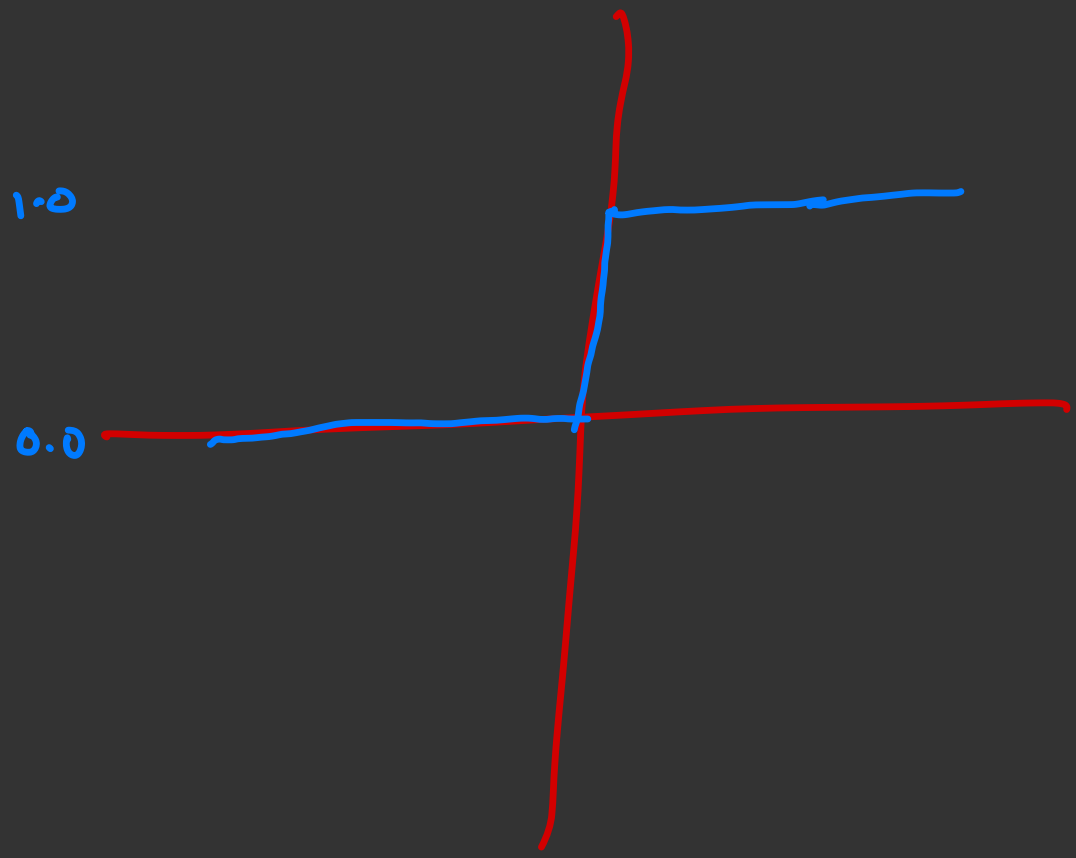
we use unit step function i.e

$$g(z) = \begin{cases} 1 & \text{if } z \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

1 if z crosses a particular threshold i.e θ

0 otherwise

graph-



in this example threshold is zero

Approximation -

$$\hat{y} = g(f(w,b)) = g(w^T x + b)$$

Perceptron update rule:

for each training sample x_i :

$$w = w + \Delta w$$

$$\Delta w = \alpha \cdot (y_i - \hat{y}_i) \cdot x_i$$

$\alpha \rightarrow$ learning rate $[0, 1)$

eg- for a 2 class problem

y	$\hat{y}$	$y - \hat{y}$	
1	1	0	(optimal)
1	0	1	(weights are low)
0	0	0	
0	1	-1	(weights are high)

* Weights are pushed towards positive or negative target in case of misclassification